Universidad Nacional de Loja

Análisis y Diseño de Software

Introducción a GitHub

Martha Suntaxi
Johanna Paz
Victor Jumbo
Andre Montoya
Wilson Iriarte

Ingeniería en Sistemas

25 de noviembre de 2015

Agenda

- Introducción
  - ¿Qué es GitHub?
  - ¿Para que sirve?
  - ¿Qué uso le daremos?
  - ¿Qué herramientas proporciona?
- Git
  - Mas sobre Git
  - Los entresijos internos de Git
  - Colaborar varias personas en un mismo Github
- Comandos Principales
  - Comandos Básicos
  - Trabajos con Ramas
  - Acciones con Archivos
- Conclusiones

UNL

¿Qué es GitHub?

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git.

¿Para que sirve?

- Aloja tu repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto.
- Se Puede contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls.

Realizar un fork es simplemente clonar un repositorio ajeno (genera una copia en tu cuenta), para eliminar algún bug o modificar cosas de él. Una vez realizadas tus modificaciones puedes enviar un pull al dueño del proyecto. Éste podrá analizar los cambios que has realizado fácilmente, y si considera interesante tu contribución, adjuntarlo con el repositorio original.

¿Qué uso le daremos?

En nuestra especialidad "Programación", fuimos aprendiendo cosas y creando programas de código abierto, fomentando el software libre. Podremos crear una cuenta gratuita y comenzar a subir repositorios de código (o crearlos desde 0), para que con la ayuda de todos ese proyecto mejore; así como también fortalecer los proyectos de los demás para crecer como grupo.

¿Qué herramientas proporciona?

En la actualidad, GitHub es mucho más que un servicio de alojamiento de código. Además de éste, se ofrecen varias herramientas útiles para el trabajo en equipo. Entre ellas, caben destacar:

- Una wiki para el mantenimiento de las distintas versiones de las páginas.
- Un sistema de seguimiento de problemas que permiten a los miembros de tu equipo detallar un problema con tu software o una sugerencia que deseen hacer.
- Una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero y debatir sobre determinados cambios realizados en un commit específico.
- Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.

GIT - SOFTWARE DE CONTROL DE VERSIONES

- ¿Qué es un software de control de versiones?
- ¿Por qué git sobre otros software de control de versiones?
- ¿Hay mas hostings que soporten git?

El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante.

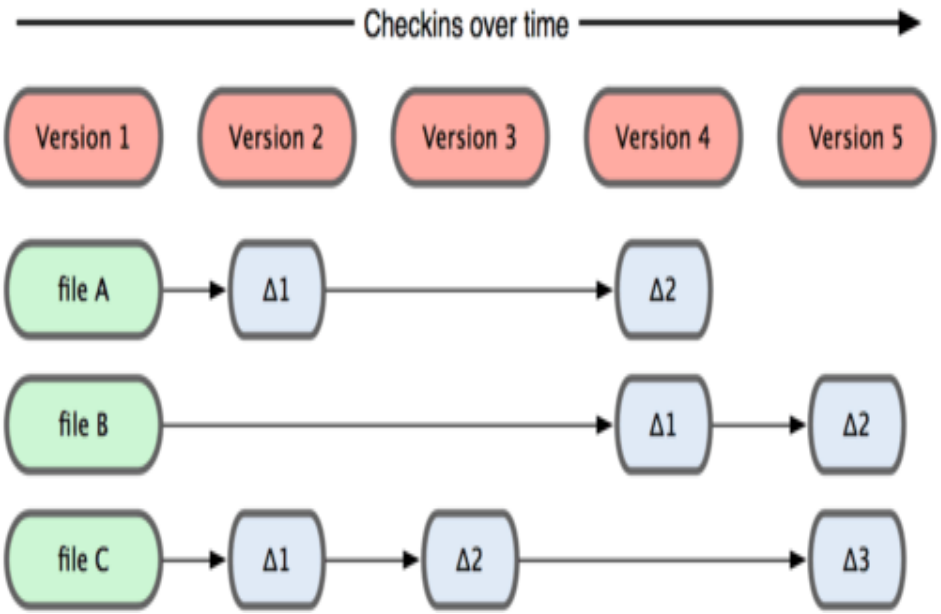
- Fácil de usar.
- Rápido y eficiente en grandes proyectos
- Sistema de ramificación (branching) para desarrollo no lineal.

Git

¿Hay mas hostings que soporten git?

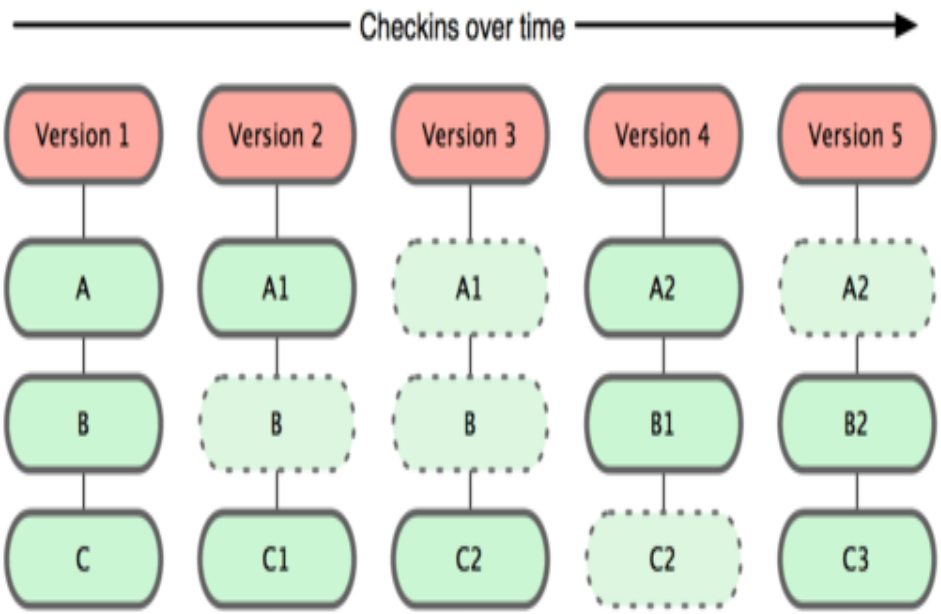
- Google Code.
- Gforce
- Assembla

Modelado de datos en Git.



Mas sobre Git

Git modela sus datos más como un conjunto de instantáneas de un mini sistema de archivos.



Mas sobre Git

- **Casi cualquier operación es local**
  - No se necesita información
  - Historia del proyecto
  - No conexión
- **Tiene integridad**
  - Todo en Git es verificado
  - Si git no lo sabe
  - El valor hash de git
- **Generalmente sólo añade información**

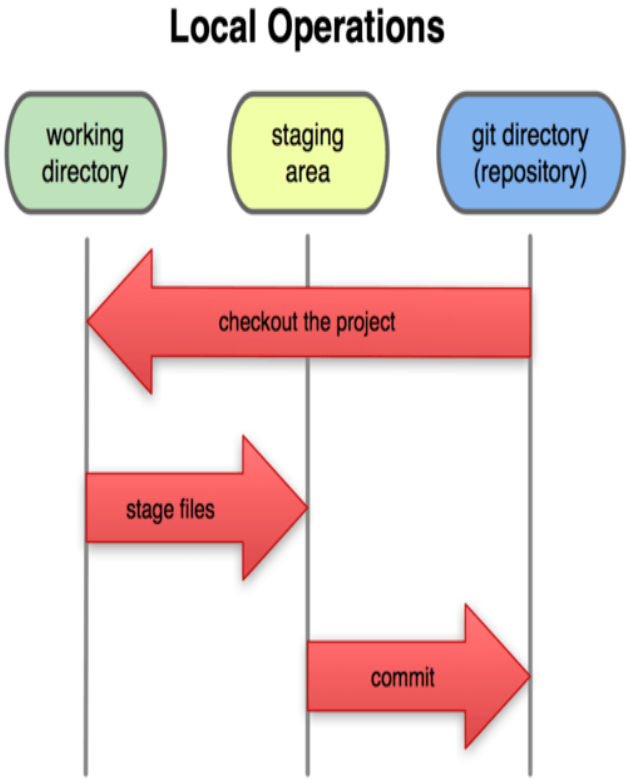
Nota: El mecanismo de comprobación de git se conoce como hash SHA-1 y es asi:

`24b9da6552252987aa493b52f8696cd6d3b00373`

UNL

Los tres estados.

- Confirmado
- Modificado
- Preparado



Los entresijos internos de Git

- ¿Qué pasa cuando lanzas 'git init'?.

```
$ ls
HEAD
branches/
config
description
hooks/
index
info/
objects/
refs/
```

Los entresijos internos de Git

Los objetos Git

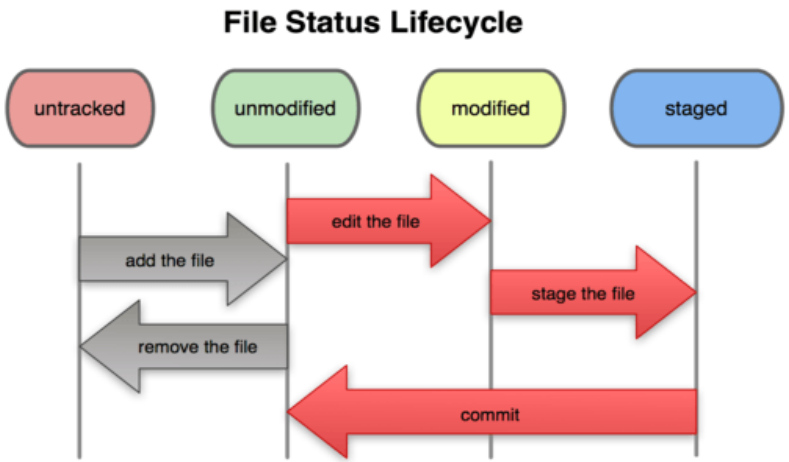
Git es un sistema de archivo orientado a contenidos.

Estupendo. Y eso, ¿qué significa?

- Inicializando un repositorio en un directorio existente `git add`
- Clonando un repositorio existente `git clone[url]`
- Sistema de ramificación (branching) para desarrollo no lineal.

Recuerda que cada archivo de tu directorio de trabajo puede estar en uno de estos dos estados:

- Bajo seguimiento (tracked).
- Sin seguimiento (untracked).



Comprobando el estado de tus archivos

```
$ git status

# On branch master

nothing to commit, working directory clean
```

Al añadir un nuevo archivo a tu proyecto

```
$ vim README

$ git status

# On branch master

# Untracked files:

#   (use "git add <file>..." to include in what will be committed)

#

#   README

nothing added to commit but untracked files present (use "git add"
to track)
```

Los entresijos internos de Git

Retornar a una version anterior

La base de datos contendrá todas versiones del archivo

```
$ find .git/objects -type f
.git/objects/1f/7a7a472abf3dd9643fd615f6da379c4acb3e3a
.git/objects/83/baae61804e65cc73a7201a7252750c76066a30
.git/objects/d6/70460b4b4aece5915caf5c68d12f560a9fe3e4
```

Para revertir el archivo a su primera versión:

```
$ git cat-file -p 83baae61804e65cc73a7201a7252750c76066a30
```

UNL

Los entresijos internos de Git

Preparando archivos modificados Esto significa que un archivo bajo seguimiento ha sido modificado en el directorio de trabajo, pero no ha sido preparado todavía, Para prepararlo, ejecuta el comando
`git add`

Los entresijos internos de Git

Confirmando tus cambios Puedes confirmar los cambios escribiendo

`git commit`

Podemos anadir la opción –v(se añadan también las diferencias de tus cambios) o –m(escribir tu mensaje de confirmación)

- –v Se añaden las diferencias de tus cambios
- –m Escribir tu mensaje de confirmación

Los entresijos internos de Git

Enviando a tus repositorios remotos `git push [nombre-remoto] [nombre-rama]`

Este comando funciona únicamente si has clonado de un servidor en el que tienes permiso de escritura, y nadie ha enviado información mientras tanto.

¿Si tú y otra persona clonan a la vez, y él envía su información y luego envías tú?

Pues sencillamente quien llegue de segundo será rechazado.

Nota: Si desean profundizar mas en el tema la guía de git en el capítulo 9 tiene mas información.

Colaborar varias personas en un mismo Github

- ¿Cómo resuelve git el problema?
- ¿Para qué sirve hacer un Fork?
- ¿Qué ventajas trae este proceso?

UNL

Comandos Básicos

- Inicializar repositorio git

```
git init
```

- Creación de un fichero README

```
vi README.md
```

- Añadir a Git los ficheros modificados.

```
git add README.md
```

- Enviar los cambios hacia GitHub

```
git push origin master
```

Ramas

Las ramas son utilizadas para desarrollar funcionalidades aisladas unas de otras.

- Crear una rama

```
git branch <nombre rama>
```

- Cambiar de rama

```
git checkout <nombre rama>
```

- Eliminar una rama

```
git branch -d <nombre rama>
```

- Renombrar una rama

```
git branch -m <vieja rama> <nueva rama>
```

Acciones con Archivos

- **Deshacer cambios locales (reset)**: Con este comando descartamos los cambios locales y volvemos al estado que teníamos guardado en el respositorio

```
git reset --hard
```

- Desahacer los cambios realizados en todos los archivos

```
git checkout -- .
```

- Actualizar repositorio local

```
git pull
```

- Fusionar ramas

```
git merge <branch>
```

- Crear una nueva etiqueta en este caso se llama 1.0.0

```
git tag 1.0.0 1b2e1d63ff
```

- Obtener el commit id

```
git log
```

Conclusiones

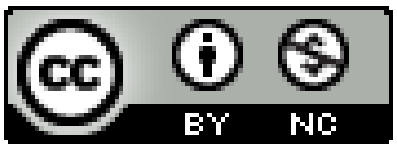
- Es un excelente repositorio para el desrrollo en grupo.
- Podemos obtener conocimientos gracias a que es un repositorio donde se encuentra codigo profesional de manera gratuita.
- Podemos colaborar con el crecimeinto de sofware libre, subiendo nuestros codigos al repositorio.

Bibliografia

1. L. Castillo, Conociendo GitHub. https://conociendogithub.readthedocs.org/en/latest/.
2. Y. Torregrosa, Álvaro.Luján Mora, Sergio,2013 http://rua.ua.es/dspace/handle/10045/26796.
3. Ruiz, José Arístides Valencia.Revista San Gregorio,2015 http://revista.sangregorio.edu.ec/index.php/RSANG/article/view/66.

UNL

Licencia



UNL

Muchas Gracias